

AD-A068 159

FEDERAL COBOL COMPILER TESTING SERVICE WASHINGTON D C  
FORTRAN COMPILER VALIDATION SUMMARY REPORT. DECSYSTEM 20. FORTR--ETC(U)  
APR 79

F/G 9/2

UNCLASSIFIED

FCCTS/FCVS66-VSR175

NL

1 OF 1  
AD  
A068159



END  
DATE  
FILMED  
6 --79  
DDC



# VALIDATION SUMMARY REPORT

FEDERAL

COBOL

COMPILER

TESTING

SERVICE



LEVEL II

(1)

## DISTRIBUTION STATEMENT A

Approved for public release  
Distribution Unlimited

Department of the Navy  
(ADPESO)

Washington, D.C.

20376

REPRODUCED BY  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
U. S. DEPARTMENT OF COMMERCE  
SPRINGFIELD, VA. 22161

MAY 3 1979

DDC  
RECEIVED  
A

DDC FILE COPY

ADA068159

AD A068159

DDC FILE COPY

1

6  
FORTRAN COMPILER  
VALIDATION SUMMARY REPORT.

14 FCCTS/L  
VALIDATION NUMBER FCVS66-VSR375

DECSYSTEM 20.  
FORTRA, Version 5A.

12 16p.

11 17 Apr 79

Prepared By:

FEDERAL COBOL COMPILER TESTING SERVICE  
DEPARTMENT OF THE NAVY  
WASHINGTON, D.C. 20376

DDC  
RECEIVED  
MAY 3 1979  
A

DISTRIBUTION STATEMENT A

Approved for public release  
Distribution Unlimited

408 353

408 438 79 05 03 02 9

<b>REPORT DOCUMENTATION PAGE</b>		1. REPORT NO. FCVS66-VSR375	2.	3. Recipient's Accession No.
4. Title and Subtitle Validation Summary Report # FCVS66-VSR375 DECSYSTEM - 20, FORTRAN 5A				5. Report Date 17 April 1979
7. Author(s) Same as organization - see 9.				6.
9. Performing Organization Name and Address Federal Compiler Testing Center (CFT) Automated Data & Telecommunications Service General Services Administration Washington, D.C. 20406				8. Performing Organization Rept. No.
12. Sponsoring Organization Name and Address Automated Data & Telecommunications Service General Services Administration Washington, D.C. 20406				10. Project/Task/Work Unit No.
				11. Contract(C) or Grant(G) No. (C) (G)
				13. Type of Report & Period Covered
15. Supplementary Notes				14.
16. Abstract (Limit: 200 words) This Validation Summary Report (VSR) for the <u>DECSYSTEM 20</u> <u>FORTRAN</u> Compiler Version <u>5A</u> ( <u>TOPS-20 MONITOR</u> Version <u>3A</u> ) provides a consolidated summary of the results obtained from the validation of the subject compiler against the 1966 FORTRAN Standard (X3.9-1966). The VSR is made up of several sections showing the discrepancies found. These include an overview of the validation which lists all categories of discrepancies; a section relating the categories of discrepancies to the language; and a detailed listing of discrepancies together with the tests which were failed.				
17. Document Analysis a. Descriptors Programming Languages      Proving Program Correctness Standards                      Software Engineering Compilers FORTRAN Verifying b. Identifiers/Open-Ended Terms FCVS CVS c. COSATI Field/Group				
18. Availability Statement  Release Unlimited		19. Security Class (This Report) UNCLASSIFIED		21. No. of Pages 18
		20. Security Class (This Page) UNCLASSIFIED		22. Price PCE 03 MFE 03

ACCESSION BY	
NTIS	NTIS SECTION <input checked="" type="checkbox"/>
DOC	Ref. Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY ORDER	
Dist.	Avail. and/or Special
A	

79 05: 03 02 9

# FORTRAN COMPILER VALIDATION -----

1. Validation Number	FCVS66-VSR375
2. Vendor	Digital Equipment Corporation
3. Mainframe	DECSYSTEM 20, Model KL2050B
4. Compiler Identification	FORTRA, Version 5A
5. Operating System Identification	TOPS-20 Monitor 3A
6. Compiler Validation System Version Number	FCVS66 1.2

\*PLEASE NOTE. The Department of the Navy may make full and free public disclosure of the Validation Summary Report (VSR) in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of this validation are only for the purpose of satisfying United States Government requirements, and apply only to the Computer System, Operating System release, and compiler version identified in the VSR. The FORTRAN Compiler Validation System is used to determine, insofar as is practical, the degree to which the subject compiler conforms to American Standard FORTRAN, X3.9-1966. Thus, the VSR is necessarily discretionary and judgmental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in the VSR are accurate or complete. The VSR is not meant to be used for the purpose of publicizing the findings summarized therein.

For information concerning this compiler you can contact the vendor's designated representative named below:

Ms. Susan Godsell  
Digital Equipment Corporation  
200 Forest Street  
Marlboro, MA 01752



## TABLE OF CONTENTS

1. SECTION 1. INTRODUCTION	1
1.1 Purpose of the Validation Summary Report	1
1.2 Preparation of the VSR	1
1.3 Organization of the VSR	1
1.4 Use of the VSR	2
1.5 Sources of Additional Information	2
2. SECTION 2. DETAILED EVALUATION OF ERRORS	3
2.1 Syntactical Errors	3
2.2 Semantic Errors	3
3. SECTION 3. SOFTWARE ENVIRONMENT	4
3.1 Compiler Options	4
3.2 Logical Unit/Physical Device Assignments	4
3.3 Compiler Optimization	6
3.4 Compiler Identification	6
3.5 Operating System Identification	6
A. APPENDIX. VALIDATION SUMMARY WORKING DOCUMENT	7
A.1 Validation Environment	8
A.2 FCVS Executive Routine Control Cards	9
A.3 Control Cards for Running FCVS Programs	11
A.4 Run Summaries	12

---

## 1. SECTION 1. INTRODUCTION

### 1.1 Purpose of the Validation Summary Report

The purpose of the Validation Summary Report (VSR) is to identify individual FORTRAN language elements whose implementation does not conform to the language specifications defined in American Standard FORTRAN, X3.9-1966.

### 1.2 Preparation of the VSR

The Validation Summary Report is prepared by analyzing the results of running the FORTRAN Compiler Validation System (FCVS). The FORTRAN Compiler Validation System consists of audit routines containing features of American Standard FORTRAN, their related data, and an Executive Routine which prepares the audit routines for compilation. Each audit routine is a FORTRAN program which includes many tests and supporting procedures indicating the result of the tests.

The testing of a compiler in a particular hardware/operating system environment is accomplished by compiling and executing each audit routine. The report produced by each routine tells whether the compiler passed or failed the tests in the routine. If the compiler rejects some language elements by terminating compilation, giving fatal diagnostic messages, or terminating execution abnormally, then the test containing the code the compiler was unable to process is deleted. The audit routine is compiled again and execution is repeated.

The compilation listings and the output reports of the audit routines constitute the raw data from which the members of the Federal COBOL Compiler Testing Service produce a Validation Summary Report.

### 1.3 Organization of the VSR

The Validation Summary Report is made up of several sections whose contents are described below.

a. Section 2 summarizes the results of the compilation and execution of the programs comprising the FORTRAN Compiler Validation System. Section 2 is divided into a subsection describing the syntax errors encountered while compiling the FORTRAN audit routines, and a subsection describing the semantic errors which occurred during execution of the FORTRAN audit routines.

b. Section 3 contains information which describes the software environment in which the compiler was tested. This includes the name and version of the operating system and the logical unit/physical device assignments used in the programs comprising the FCVS. The options used with the compiler are also given, and if applicable, the use of compiler optimization features is explained.



---

c. Appendix A is the Validation Summary Working Document, a working paper resulting from the compilation and execution of the FCVS. The VSR is derived from Appendix A.

#### 1.4 Use of the VSR

The Department of the Navy may make full and free public disclosure of the Validation Summary Report (VSR) in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of the validation are only for the purposes of satisfying United States Government requirements, and apply only to the computer system, operating system release, and compiler version identified in the VSR.

The FORTRAN Compiler Validation System is used to determine, insofar as is practical, the degree to which the subject compiler conforms to the FORTRAN Standard. Thus, the VSR is necessarily discretionary and judgmental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in the VSR are accurate or complete. The VSR is not meant to be used for the purpose of publicizing the findings summarized therein.

#### 1.5 Sources of Additional Information

The detailed FORTRAN language specifications are given in the publication "American Standard FORTRAN, X3.9-1966", available from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

An explanation of the FORTRAN Compiler Validation System is contained in the FCVS User's Guide. This document explains how to run the compiler validation system. The User's Guide and a magnetic tape containing a copy of the FCVS programs are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia, 22151. (Ordering information can be obtained from the Federal COBOL Compiler Testing Service.)

## 2. SECTION 2. DETAILED EVALUATION OF ERRORS

This section summarizes the results of the compilation and execution of the programs comprising the FORTRAN Compiler Validation System (FCVS). The version of the FCVS used during this validation is shown inside the front cover of the VSR.

Section 2 is made up of two subsections. The first subsection describes each syntax error encountered during compilation of the audit routines, and the second subsection describes the semantic errors encountered during execution of the audit routines.

Each error or deviation noted in this section makes reference to a program contained in Appendix A (Validation Summary Working Document). This reference provides the documented results of an occurrence of errors/deviations detected during the running of the FCVS using the compiler within the environment identified in this document. The Validation Summary Working Document is presented in sequence by program number.

### 2.1 Syntactical Errors

No errors.

### 2.2 Semantic Errors

No errors.

---

### 3. SECTION 3. SOFTWARE ENVIRONMENT

The compiler referenced in this document was validated using the software environment described in this section. When using a modification of the described environment, the compiler may or may not continue to conform to the Standard. It should be noted that during the validation process, an attempt is made to validate as many different options as possible.

The use of compiler options, logical unit/physical device assignments, and any form of optimization which is not described in this report could cause the compiler to produce a program that does not perform according to the specifications of Standard FORTRAN. Only the environment described in this document has been used with this compiler to satisfy the validation requirements of the Department of the Navy.

#### 3.1 Compiler Options

The options or parameters used on the processor call statement for the compiler are listed as follows:

- a. Batch - Without Optimization.

EXECUTE/COMPILE/FORTRAN/LIST FMnnn.NFT

- b. Batch - With Optimization.

EXECUTE/COMPILE/FORTRAN/LIST/OPTIMIZE FMnnn.NFT

- c. Time-sharing - Without Optimization.

EXECUTE/COMPILE/FORTRAN/LIST FMnnn.NFT

- d. Time-sharing - With Optimization.

EXECUTE/COMPILE/FORTRAN/LIST/OPTIMIZE FMnnn.NFT

#### 3.2 Logical Unit/Physical Device Assignments

Logical units were specified by inserting OPEN statements in each program with the required file specifications and then making the physical device assignments at runtime. A file called NAVDEV.CMD contained the initial (default) device assignments. When a change in assignment was necessary, such as in modifying FM100-FM108 from mass-storage to tape, the change was made within the NAVDEV.CMD file.

- a. Batch Compiler Runs.

Printer Destined Files:

---

---

Logical unit 06 was assigned to the system printer by using the following OPEN statement within the source program:

```
OPEN(UNIT=I02,DEVICE='OUT',FILE='FMnnn.LPT')
```

At runtime the logical unit was assigned to a physical device as follows:

```
@DEFINE OUT: DSK:
```

Tape Files:

Logical unit 07 was assigned to a tape file by using the following OPEN statement within the source program:

```
OPEN(UNIT=I06,DEVICE='DAT',FILE='FMnnn.Dnn')
```

At runtime the logical unit was assigned to a physical device as follows:

```
@TMOUNT DAT: NAVYWK,  
@WRITE-ENABLED
```

Sequential Mass-storage Files:

Logical unit 07 was assigned to a mass-storage device by using the following OPEN statement within the source program:

```
OPEN(UNIT=I06,DEVICE='DAT',FILE='FMnnn.Dnn')
```

At runtime the logical unit was assigned to a physical device as follows:

```
@DEFINE DAT: DSK:
```

Card Input Files:

Logical unit 05 was assigned to the card input file by using the following OPEN statement within the FM015 source program:

```
OPEN(UNIT=I01,DEVICE='IN',FILE='FM015.NFD',ACCESS='SEQIN')
```

At runtime the logical unit was assigned to a mass-storage file which contained the FM015 data card images:

```
@DEFINE IN: DSK:
```



b. Time-sharing Compiler Runs.

The time-sharing runs were submitted in the same manner as the batch runs. All logical and physical device assignments remained the same with the exception that input to FM015 was entered from the terminal and output from FM015 was displayed on the terminal via the following commands:

```
@DEF IN: TTY:
@DEF OUT: TTY:
```

3.3 Compiler Optimization

The compiler may or may not have optimization features. If there was an optimization feature available, it was used during the validation process (during a separate execution of the Compiler Validation System) to determine if its use causes the compiler to produce a program which does not give the expected results. If the optimization is invoked through the compiler call statement then it is mentioned in paragraph 1 above. If it is invoked through the introduction of a compiler directing source program statement, it is shown below. Optimization which would require modification to source program statements is not considered in this report in that it is beyond the scope of the use of Standard FORTRAN and the validation process.

The optimization feature for this compiler is invoked by specifying the word OPTIMIZE as an option on the EXECUTE statement for the FORTRAN compiler. See Section 3.1 above.

3.4 Compiler Identification

FORTRA, Version 5A

3.5 Operating System Identification

TOPS-20 Monitor 3A



---

A. APPENDIX. VALIDATION SUMMARY WORKING DOCUMENT

This appendix is a working paper produced during the validation and documents the results of the compilation and execution of each of the programs comprising the FCVS. The results contained herein are based on the use of the compiler within the Validation Environment identified in this appendix. This appendix (Validation Summary Working Document) is not part of the official Validation Summary Report (VSR), and it is not intended to reflect in any way the compiler's usefulness or degree of conformance to the language specifications.

The reader of this appendix should keep in mind that the same problem area may appear in more than one program but is considered only as a single discrepancy, and the problem is reflected only once in the body of the VSR. (The VSR will in turn only reference the first occurrence of the problem in the appendix.)

This appendix is divided into four parts. The first part describes the Validation Environment. The second part lists the Monitor Input Cards used in creating a job control stream for execution in the batch mode. The third part shows the control cards required to compile and execute an individual program. The fourth part of the document is divided into two categories of information: compilation results and execution results. Information items, such as compiler warning messages, are included in the summary of compilation and execution results.

The reference document for FORTRAN is American Standard FORTRAN, X3.9-1966.

## A.1 Validation Environment

COMPILER IDENTIFICATION:	FORTRA, Version 5A
COMPUTER SYSTEM:	DECSYSTEM 20, Model KL2050B
OPERATING SYSTEM:	TOPS-20 Monitor 3A

Each of the programs comprising the FCVS was compiled and executed twice in the batch mode. First, the programs were compiled and executed without the compiler optimization feature, and then the programs were compiled and executed with the compiler optimization feature requested. In addition, the programs which test I/O, FM100 through FM108, were run with two different I/O devices. The programs were first run with the output logical unit assigned to a tape device, and then rerun with the unit assigned to a mass-storage device. Upon completion of the batch runs the entire FCVS was rerun in time-sharing mode in the same manner as described for the batch runs.

## A.2 FCVS Executive Routine Control Cards

The FCVS Executive Routine was used to prepare the FCVS programs for execution in the batch mode. The Control Cards used as input to the FCVS Executive Routine together with the job control stream for executing the Executive Routine are listed below:

```

@ASSIGN MT632:
@DEFINE X: MT632:
@RUN NAVTAP.FOR      (Copy FCVS population file from tape to disk)
@RUN SNOBOL           (Verify that the file copy was successful)
*SCNNVY
.
.
@UNL X:
@DEAS X:
@RUN SNOBOL           (Create input control file for FCVS executive
*MAKINP               program in file MYEXEC.INP. The contents of this
.                     file are as follows:)
.
PFM001,PFM109
*LIST      U
*END-MONITOR
*BEGIN-UPDATE
*START,FM001
=00590
      OPEN(UNIT=IO2,DEVICE='OUT',FILE='FM001.LPT')
.
.
*START,FM015
=00740
      OPEN(UNIT=IO1,DEVICE='IN' ,FILE='FM015.NFD',ACCESS='SEQIN')
=00770
      OPEN(UNIT=IO2,DEVICE='OUT',FILE='FM015.LPT')
.
.
*START,FM100
=00930
      OPEN(UNIT=IO2,DEVICE='OUT',FILE='FM100.LPT')
=01170
      OPEN(UNIT=IO6,DEVICE='DAT',FILE='FM100.D06')
.
.
*END-UPDATE
*END-INPUT
.
.

```

---

@RUN MYEXEC

(Compile and execute the FCVS executive program.  
Build individual files for each audit routine.)

@PRI MYEXEC.INP,NFT2\*.CTL

@PRI MYEXEC.LPT/FILE:FORTTRAN

@SU NFT2N.CTL/TIME:00:30:00 (Submit batch job.)



---

### A.3 Control Cards for Running FCVS Programs

The job control stream for an individual program consisted of the following control cards:

For programs FM001-FM014, FM016-FM099 and FM109:

```
@EXECUTE/COMPILE/FORTRAN/LIST  FMnnn.NFT
```

(audit routine source program in file FMnnn.NFT)

```
@RENAME FMnnn.REL  FMnnn.REU
```

```
@RENAME FMnnn.LPT  FMnnn.LPU
```

For program FM015:

```
@EXECUTE/COMPILE/FORTRAN/LIST  FM015.NFT
```

```
*G
```

```
*G
```

```
@RENAME FM015.REL  FM015.REU
```

```
@RENAME FM015.LPT  FM015.LPU
```

For programs FM100-FM108 with mass-storage files:

```
@EXECUTE/COMPILE/FORTRAN/LIST  FMnnn.NFT
```

```
@RENAME FMnnn.REL  FMnnn.REU
```

```
@RENAME FMnnn.LPT  FMnnn.LPU
```

```
@RENAME FMnnn.Dnn  FMnnn.Unn
```

For programs FM100-FM108 with tape files:

```
@TMOUNT  DAT: NAVWRK,
```

```
@WRITE-ENABLED
```

```
@EXECUTE/COMPILE/FORTRAN/LIST  FMnnn.NFT
```

```
@RENAME FMnnn.REL  FMnnn.RTU
```

```
@RENAME FMnnn.LPT  FMnnn.LPU
```

```
@REWIND DAT:
```

```
@COPY DAT:  FMnnn.Tnn
```

```
@REWIND DAT:
```



## A.4 Run Summaries

## FM001 through FM045

## A. Compilation

No errors.

## B. Execution

No errors.

## FM050 through FM062, FM080 through FM083, and FM097 through FM099

## A. Compilation

No errors.

## B. Execution

No errors.

## FORTRAN I/O Routines - FM100 through FM108

These audit routines were executed with the output logical unit assigned to a tape device and rerun with the output logical unit assigned to a mass-storage device.

## FM100 through FM108

## A. Compilation

No errors.

## B. Execution

No errors.

## FM109

## A. Compilation

No errors.

## B. Execution

No errors.